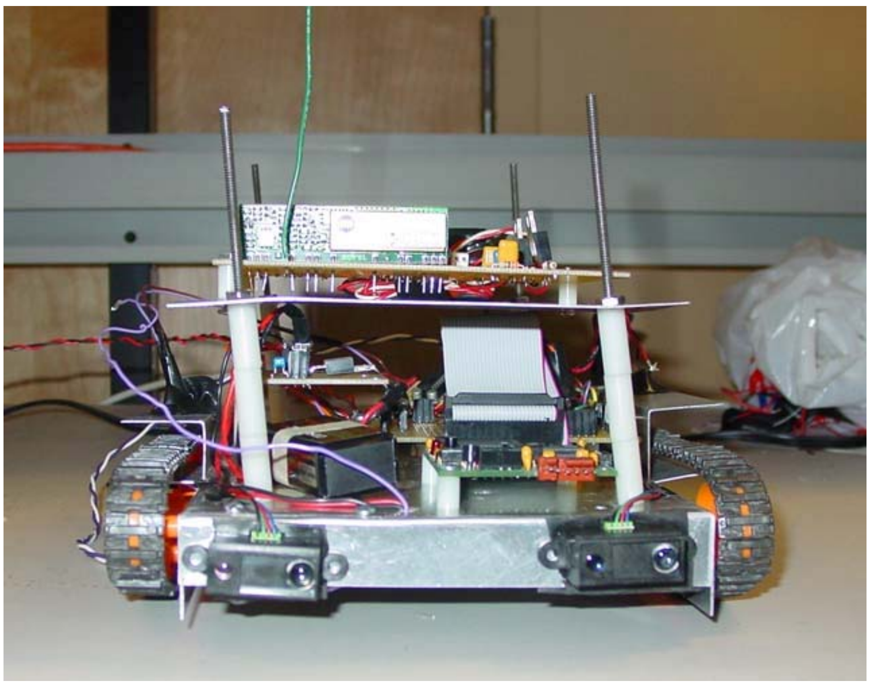
THE WATCHDOG

Submitted By:
Group 1

EECE 474
The University of British Columbia
Electrical and Computer Engineering Department

THE WATCHDOG



Submitted to:
Dr. W. G. Dunford
November 17, 2003

By:
Katie Man Ki Au 43396001
Murtaza Ibrahim Bachoo 4238400
Amin Eskooch 76789999
Peter Gulyas ()
Xiongfei Meng 75307017
Granaz Ziaghazvini 93884997

ABSTRACT

"The WatchDog"

The purpose of the WatchDog is to provide manual inspection of an area that is prone to intruders and possible fire hazards. Two sensors are mounted on the WatchDog to detect humans. One detects the presence of heat source through infrared radiation and the other concentrates and maximizes this detection. A temperature sensor is implemented in order to warn an inspector of the possible occurrence of fire. The WatchDog is able to detect the presence of obstacles in the navigation path due to distance measuring sensors. In order to facilitate the navigation of the WatchDog, optical encoders are also added. The WatchDog is able to alert an outside source when an intruder is detected through RF communication. All the comprising components of the WatchDog are mounted on an aluminum chassis with treads. This paper precisely describes each component and the system as a whole in detail.

TABLE OF CONTENTS

LIST OF TABLES AND ILLUSTRATIONS

TABLES

FIGURES

GLOSSARY

Application Program Interface	The interface (calling conventions) by which an application program accesses operating system and other services. An API is defined at source code level and provides a level of abstraction between the application and the kernel (or other privileged utilities) to ensure the portability of the code.
ASCII	A code for information exchange between computers made by different companies; a string of 7 binary digits represents each character; used in most microcomputers
Bluetooth	A specification for short-range radio links between mobile computers, mobile phones, digital cameras, and other portable devices
Child	A Multiple Document Interface form. Each child form contains a single document type that the programmer defines using controls.
DTR	The wire in a full RS-232 connection that tells the Data Communication Equipment (DCE, typically a modem) that the Data Terminal Equipment (DTE, typically a computer or terminal) is ready to transmit and receive data
DLL	A file containing a collection of Windows functions designed to perform a specific class of operations. Functions within DLLs are called by applications as necessary to perform the desired operation
Duplex	Allowing communication in opposite directions simultaneously
IEEE 1394	A 1995 Macintosh/IBM PC serial bus interface standard offering high-speed communications and isochronous real-time data services
Form	An essential module that stores the main functions of VB source codes

Pyroelectric	Pertaining to, or dependent on receiving electric polarity when heated.
Radio Frequency	An electromagnetic wave frequency between audio and infrared
RS-232	The most common asynchronous serial line standard that specifies the gender and pin use of connectors (also known as EIA-232)
Serial Port	An interface (commonly used for modems and mice and some printers) that transmits data a bit at a time
TTL	A common semiconductor technology for building discrete digital logic integrated circuits. It originated from Texas Instruments in 1965
Universal Serial Bus	An external peripheral interface standard for communication between a computer and external peripherals over an inexpensive cable using biserial transmission.
Visible	A state that defines if a control or form is accessible by other control or forms

LIST OF ABBREVIATIONS

API .... Application Program Interface
ASCII .... American Standard Code for Information Interchange
CMOS .... Complementary Metal Oxide Semiconductor
CTS .... Clear To Send
DLL .... Dynamic Link Library
DTR... Data Terminal Ready
DSR.. Dynamic Service Register
DVC .... Digital Video Creator
FET .... Field Effect Transistor
FSM... Finite State Machine
IR .... Infrared
LED... Light Emitting Diode
PIR .... Pyroelectric Infrared
NTSC .... National Television Standards Committee
RDRF .... Receive Data Register Flag
RF.. Radio frequency
RIE .... Receive Interrupt Enable
RTS .... Request To Send
SCDR .... Serial Communication Data Register
TTL .... Transistor-Transistor Logic
USB.. Universal Serial Bus
VB .... Visual Basic

1.0 INTRODUCTION

There has always been the need for security in important and sensitive locations that are prone to attacks by intruders. This demand has caused a mass production of security systems such as alarm systems and surveillance cameras. In addition to such systems, security personnel are hired to provide sound protection against possible attackers. The WatchDog is designed to reduce and eliminate the necessity of these systems and personnel by performing some of their duties accordingly. Therefore, the WatchDog will decrease the excessive need of security personnel while ensuring a secure environment.

The objective of our project was to design a system that would scout an area of interest autonomously or manually by the operating agent. It was to be equipped with sensors to detect the presence of possible intruders and potential fire hazards. If a security breach is detected, the monitoring agent is notified immediately and the subject is prompted to identify himself with a code to confirm the breach. If the robot confirms a security breach, a security monitoring agent would be contacted. In case of a fire hazard-which would be detected through heat sensors- the agent would again be informed. Applications of this robot include homes, industrial warehouses, banks, offices, shops and virtually any place subject to intrusion or fire hazard. Our decision to implement this system was based on different factors some of which include the vast applicability of such a system and the complexity of the robot. This project allowed us to utilize our knowledge from electrical, computer and mechanical engineering fields.

Our main challenge during the project was the 2.5-month time constraint. We were therefore unable to incorporate additional features into our robot as we had anticipated. Hence we decided to complete the main sections indicated in our design documentation report and are proud to say that the majority of the design document objectives were met.

This report provides a detailed description of the hardware and software aspects of the WatchDog. It includes a fine outline of the chassis, RF* components, sensors and camera and intruder interface. It finally presents the reader with an expense chart and concludes with a summary.

*This and all subsequent terms marked with an asterisk are defined in the Glossary.

2.0 THE CHASSIS

The structure of the body is designed to function as a moving platform with the capability to turn on 90^o, 180^o, 270^o and 360^o on a single point. The body is constructed of two main decks and two sub levels that hold the different aspects of the communication, sensory equipment and power.

2.1 SHAPE AND MATERIAL

The design constraints of the shape of the robot are balance and size. The balance of the robot influences the stability of rotation and the size of the decks, whereas the size of the body determines the power consumption and moving speed. To maximize stability and size, the robot is designed to have a wide base and near equal weight distribution along the length of the body. The increase of the base size causes the center of balance to be lower so that when 90^o, 180^o, 270^o and 360^o turns are executed maximum stability is achieved. Weight distribution along the length of the body will increase the rolling friction causing the power efficiency to lower. Furthermore, aluminum is used in the construction of all parts to attain the goal of keeping the robot lightweight so that power consumption is decreased.

2.2 COMPONENTS AND CORRESPONDING FUNCTIONS

The body is contrived of two main decks, two sub levels, two tank-like treads driven by a twin dc motor and a 201:1 gear system. Deck 1 (refer to Figure 1) holds the microcontroller PCB, 1 voltage regulator PCB, shaft encoders, optical sensors, power supply, and buzzer. Sub level 1 (refer to Figure 1) houses the twin dc motor and 201:1 gear system, H-Bridge PCB, and two voltage regulator PCBs. Deck 2 (refer to Figure 1) possess the two-way radio frequency communication device PCB, human detection system structure and PCB, and keypad. Sub level 2 (refer to Figure 1) supports the cordless camera.

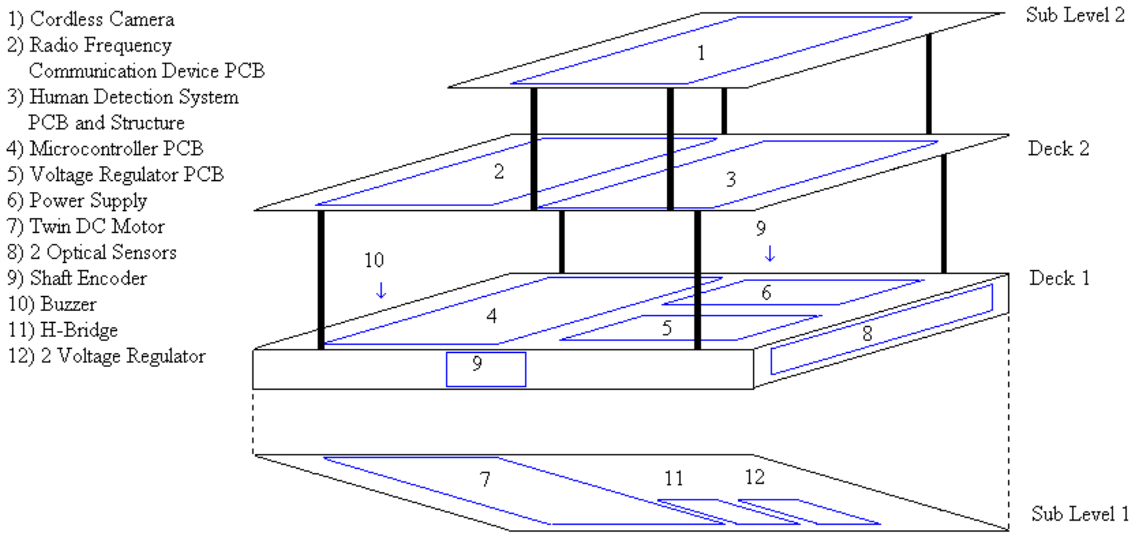


Figure1 Body Illustration

2.3 DESIGN ADVANTAGES

By having a wide base and near equal weight distribution along the length of the body design, stability and speed have increased and power consumption is reduced, compared to other wheel driven designs. Also by having tank-like treads instead of wheels other advantages are observed. The treaded robots course digresses by 45% less than the wheel driven model during a 1-minute course deviation test trial. Testing the turning capability of both robots on a 90^o, 180^o, 270^o and 360^o turns, the center of rotation displacement was measured to be approximately 0 cm on all cases for the treaded design, while the wheeled design was observed to have a deviation of between 5 to 10 cm. Testing the adaptability to different ground surfaces the wheeled robot would be obstructed on many surfaces such as carpet, tile, cement and also protruding objects for example welcome mats and the metal bar in a door way; the treaded version of the robot rolled over all of these obstructions with little to no course deviation. In addition the inherent shape of the tank treads proved to make the shaft encoding very easy to mount and implement. Also the cost of the treaded version of the robot is less because only tank-like treads and the gearbox had to be purchased, whereas the wheeled robot needed two casters, two wheels and a gearbox as well. The total difference in price is $15.00.

2.4 DESIGN DISADVANTAGES

The disadvantages to the current design come from the treads. The length of the treads requires a larger area on the body of the robot, increasing the amount of material that needs to be used, which in turn increases the weight of the robot. However, this increase of weight is minimal. Also because of the length of the treads, more parts are used in attaching the treads securely to the body; this increases assembly costs and the time spent on building the body.

3.0 THE MICROCONTROLLER

Functions of the WatchDog are monitored and controlled using MicroCore11 Controller Board by Technological Arts Corp. At the heart of MicroCore11 lies the ever popular Motorola M68HC11 E series with 32 K of External Ram and 32K of EEPROM. Figure 2 shows an up-close figure of the Microcore11 board.

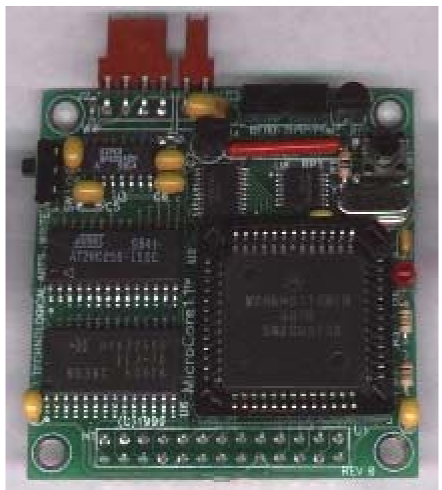


Figure2. MicroCore11 Board

MicroCores11's compact design, versatility and convenient communication setup make it an ideal candidate for applications such as robotics where space constraints and large number of I/O lines are of critical importance. Other factors in choosing MicroCore11 as the controller include availability and familiarity. Functions of the Controller can be broken down into the following:

1. Governing movements of the WatchDog via controlling the select lines into the H-Bridge used to drive the motor.
2. Initializing and maintaining communication with the PC unit.
3. Monitoring various sensors such as temperature and IR sensors installed on the WatchDog and controlling the actions of the robot accordingly.
4. Detecting and verifying the identity of a possible intruder using the human detection sensor and the keypad.
5. Computing the distance traveled by the robot using the information acquired by the optical encoder and controlling the path of motion accordingly.

3.1 MOTOR CONTROL

The driving force of the WatchDog is two small DC motors which combined with a gearbox (203:1 gear ratio) provide enough torque to move the robot in any direction. The two motors drive two individual hexagonal shafts and are therefore able to drive the robot individually as seen in the figure below.

Motors used are rated for 3 volts but they work fine up to 6 volts. However, at higher voltages they prove to be excessively noisy. Therefore, they are supplied with a maximum of 5 volts using a voltage regulator on board of the WatchDog. Figure 3 shows the gearbox.

Figure3.GearBox (203:1)

An H-bridge had to be utilized to amplify the current because of the low current drive ability of the microcontroller output lines. Figure 4 demonstrates the L298 H-Bridge Kit from HVW Technologies, used to drive the motors of the WatchDog. The H-bridge can control two motors simultaneously using only four select lines and it can output up to a maximum of 2 Amps per motor (4 Amps Total) and can take up to 50 volts. Also sharp peaks of up to 3-5 volts are tolerated.

Figure4. L298 H-bridge Motor Drive Controller Kit

The L298 uses inverse logic for select lines and regular logic for the enable lines. The 4 select lines for the 2 motors, namely:

```
Fwd(left),Back(left),Fwd(right),Back(right)
```

were tied to PA7-PA4 which are 4 output lines of the MicroCore11. In order to drive the robot in the desired direction the appropriate select lines are grounded. As an example, in order to go forward, the two forward lines (PA7 & PA5) are grounded and the backward lines (PA6 & PA4) are driven to high. Table 1 summarizes the select line values for each direction:

Table 1. Select line values for each direction

Direction	PA7	PA6	PA5	PA4
Forward	0	1	0	1
Backward	1	0	1	0
Right	1	0	0	1
Left	0	1	1	0
Stop	1	1	1	1

Further details regarding the theory of operation of the H-Bridge is beyond the extent of this paper.

3.2 COMMUNICATION

One of the essential components of the WatchDog is the RF communication module. The robot has to be constantly in contact with the remote unit. The task of communication initialization and maintenance of this communication is done through the serial interface on the MicroCore11 through PORTD on the HC11. The controller initially sets the baud rate to 300 in order to synchronize itself with the RF module. RIE Interrupts are used in order to establish communication between the RF transceiver and the HC11. In the corresponding interrupt service routine, namely, `SCIHandler(),`the value of the RDRF flag is checked, and if set the value of the SCDR is read as the received character. Security bits are also implemented in the transmission protocol in order to avoid interference and enhance the transmission security. The transmission protocol is described in more detail further sections of this report.

3.3 MONITORING

In addition to communication and motion control, the controller constantly monitors the PIR sensor and IR sensors mounted on the WatchDog. The PIR sensor, described previously, senses the presence of an intruder. The Microcontroller will then stop the robot and transmit a secure message to the PC, informing the monitoring agent of the intrusion. Before approaching the intruder, a picture of the surrounding environment is taken and is saved on the remote PC hard drive for evidence. The WatchDog will then start the verification sequence and sound its alarm. The distance sensors are used to inform the robot of a possible obstacle on its way. Once an obstacle is detected the robot will start a pre-programmed sequence to divert its path and move passed the obstacle.

The temperature sensor on board provides the WatchDog with temperature information and allow it to detect a possible fire hazard. A threshold value for the temperature is programmed in the controller and an alarm message is generated once the threshold value is exceeded.

3.4 IDENTITY VERIFICATION SEQUENCE

If the PIR sensor informs the WatchDog controller of the presence of an intruder, an external interrupt will cause the controller to stop all current operations and instead execute the verification sequence service routine. In the verification sequence routine, a message is sent to the PC unit immediately, informing the monitoring agent of the detection of a possible intruder. The monitoring agent will then decide whether to switch to manual mode, take pictures or inform the Police. If left in the automatic mode, the WatchDog will approach the intruder. The keypad installed on the robot will provide for an easy way to verify the intruder's identity. A 4-digit code will be entered and compared to the stored library of codes in the memory. If a successful match is made the WatchDog will exit the service routine and continue its normal operation. Otherwise, the WatchDog will send another message to the PC unit and sound its alarm. Figure 5 shows the flow chart of the verification sequence service routine.

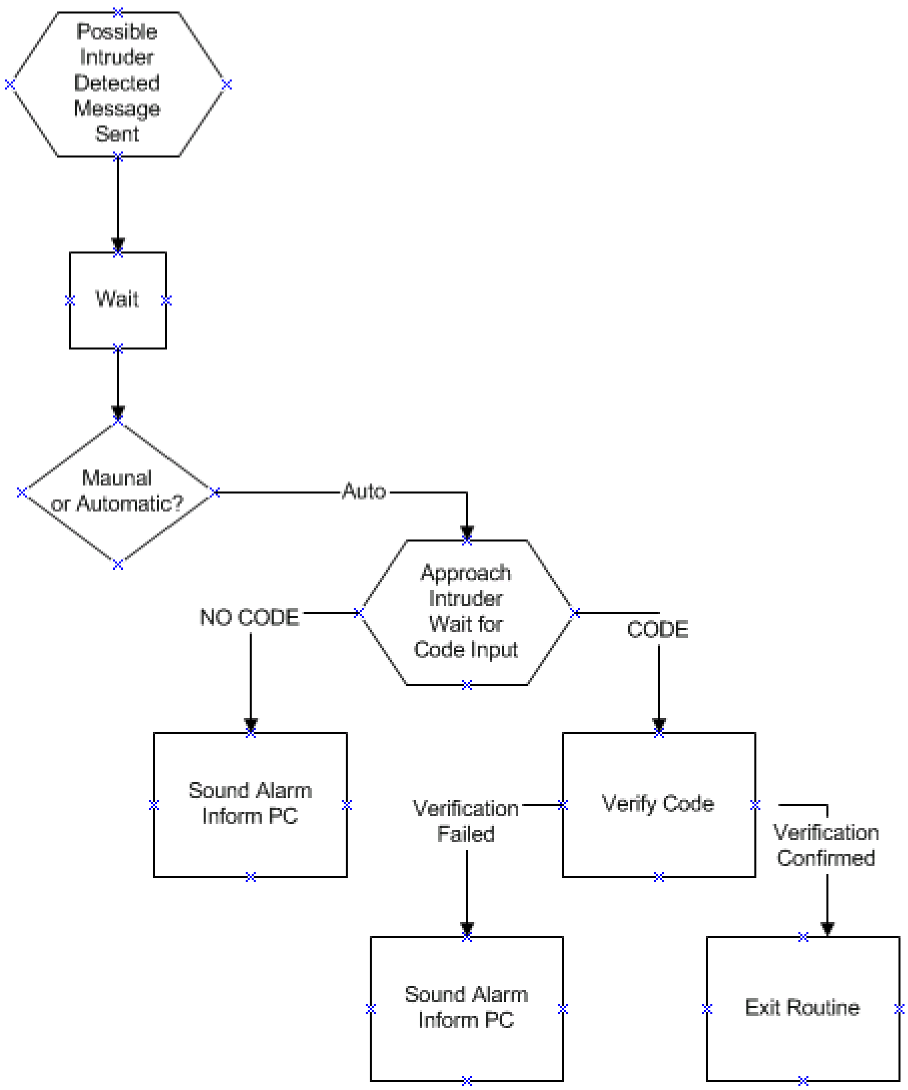


Figure5. Verification Sequence ISR

3.5 NAVIGATION

The output of the encoders is a square wave which is fed to pins PA1 and PA2. These pins are to operate in "Timer Input Capture" mode. In this mode the value of the timer input capture register is incremented every time a rising edge of a signal is detected on the corresponding pin. Thus, by counting the number of edges, the controller has a means of determining how far the robot has traveled. When in pre-program mode, the controller uses this information to navigate the robot according to the user specifications.

4.0 RADIO FREQUENCY COMMUNICATION

The following sections provide the reader with an overview of the RF Communication between the robot and the monitoring computer.

4.1 RF TRANSCEIVER SPECIFICATIONS AND SELECTION

An essential requirement of the WatchDog robot is to alert an outside source or a monitor, when an intruder is detected. The monitor then also requires the ability to instruct the WatchDog on what steps to take, and occasionally be able to alter the WatchDog's path. These requirements necessitate duplex* RF communication.

The data transfer rate and the frequency of transmission do not require a transceiver with a high baud rate. The data being transferred from both sources is at most one or two bytes at a time. The data is also transferred only when interrupts in the systems routines are required or have been generated.

Since the WatchDog is designed to operate in a warehouse-like area, the communication range is specified to at least 200 feet. This is the maximum distance that the robot can be away from the remote monitoring station. In areas or buildings where this station is central, the area of coverage would be up to 124,800 ft sq.

Also due to the robot's features as a remote unit, the power consumption of the transceiver has to be minimal. This requirement can be relaxed to state that the power consumption of the transceiver in receive mode should be minimal, and that the transceiver should be in receive mode for most of the time. Particularly, a transceiver which consumes less than 20 mA in the receive mode would be suitable.

In view of the above specifications, the RF team chose the ATXR-434-ULC transceivers available from Abacom Technologies. Figure 6 shows the front and side view of these transceivers:

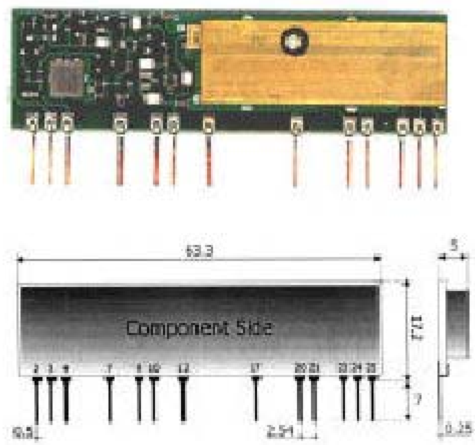


Figure6. The ATXR-434-ULC transceiver

This transceiver is capable of half duplex communication over a range of 250 ft. with a suitable antenna. Although the manufacturer does not rate the maximum transmission rate of the transceiver, it is safe to assume that this transceiver would have the required transmission rate. The transceiver is designed to be used for security monitoring according to the manufacturer’s datasheet.

This particular transceiver remains in receive mode until a high voltage is received on the transmission line. In the receive mode the transceiver utilizes 10 mA and in the transmit mode it utilizes 70 mA. These specifications also meet our power consumption requirements.

The only drawback of this transceiver is that it is rated at a maximum input voltage of 3V. Since the power supply on the robot is 9V, and most ICs on the robot will utilize 5V, the transceiver requires its own voltage regulator. A greater concern is that the transmission input and output pins are also set to 3V maximum. These ratings do not correspond to TTL* logic voltage levels. As such, the voltage output from the microprocessor and the RS-232* port will have to be lowered to 3V.

Despite the above concerns, this transceiver is the best unit to for the WatchDog robot.

4.1.1 Transceiver Hardware Design on the Robot

The data output from the microprocessor is in serial format. The Serial Communication Interface was chosen because of its low external hardware requirements and also because it can be connected to the serial pins PD1 and PD2 of the microprocessor.

The primary hardware consideration is the voltage values between the microprocessors serial pins and the transceiver's transmit and receive pins. A max232 chip (DS14C232CN) by National Semiconductors is used to convert +12/-12 V down to 0V to 5V. A voltage divider is then implemented to convert the 5V output from the chip down to 3V. The max232 chip reads a 3V input as high, and as such the output of the transceiver does not need to be converted to 5V.

The max232 chip outputs a 5V when the RS-232 input is at -12V. As this output is connected to the transmission pin of the transceiver, it effectively holds the transmitter in transmission mode. The serial output from the max232 is disconnected from the transceiver through an analog switch. An enable pin, PD2, from the HC11 is used to control the switch. The transceiver is in transmit mode for the duration for which the enable pin is held high. This allows the transceiver to receive data in the remaining time.

The analog switch and the max232 chip require a typical voltage supply of 5.5V. Therefore another voltage regulator is implemented on the transceiver PCB to provide a 5.5V supply. The transceiver PCB utilizes 10-19 mA in receive mode, and 70-79 mA in transmit mode. These voltage levels are suitable for use on the robot and at the monitor end.

The Antenna implemented on the transceiver is a ¼ wavelength, 16.5 cm, of 22 gauge single copper tinned wire. Although the option of mounting an industrial antenna exists, it is very difficult to do impedance matching and ensure that the impedances are matched. As such the gain in the communication range cannot be verified from using an antenna. Due to this added complexity we decided to utilize the copper tinned wires as our antennas.

4.1.2 Transceiver Unit Design at the Monitor End

The transceiver design at the monitor end is very similar to the design at the robot end. The major difference in the robot end was the inclusion of a 90º DB-9 connector and a power jack to power the unit from a regular socket through an adapter. The unit contains an on-board voltage regulator

and therefore the adapter is not constrained to a certain voltage. Any adapter rated 7V or higher will be suitable for the unit.
Figure 7 demonstrates the transceiver unit at the monitor end of our project:

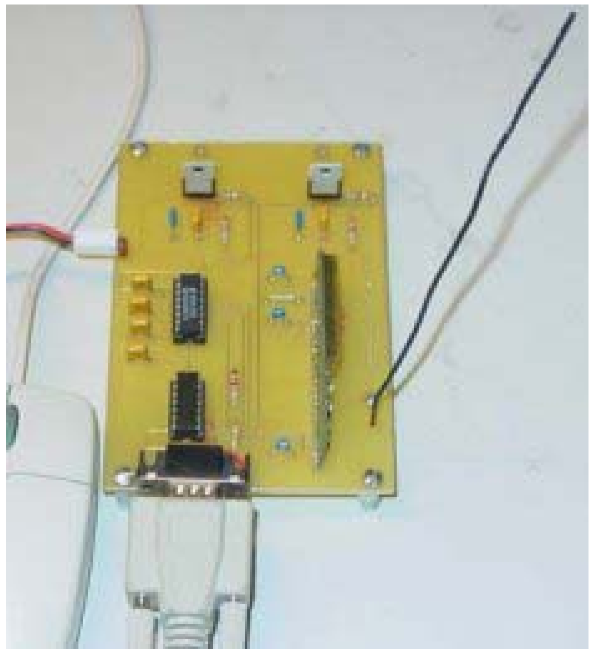


Figure7. Transceiver unit at the monitor end

The unit also has an enable pin that is controlled by the DTR*-enable pin (pin2) on the RS-232 interface. For transmission mode to be enabled, this pin must be held high for the duration of the transmission. Also due to timing considerations, this pin should be held high for a few milliseconds before the data is transmitted and brought low a few milliseconds after the transmission is complete.

The transceiver at the monitor end and at the robot end are both maintained in receive mode. The main reason for this is because when the transmitter is in transmit mode, it does not receive any data which would disable communication. Keeping the transceivers in this state also lowers power consumption.

4.2 COMMUNICATION PROTOCOL

Although the transceivers have built in noise protection, the units pick up a large amount of RF noise. This noise can be reduced by squelching or decreasing the RF gain of the transceivers. However this does not filter out all the noise. As such a communication protocol is implemented to ensure the security and integrity of the transmission.

In the RS-232 communication protocol, a start bit is included before every byte of data. This start bit is a change from low ("1") to high ("0"). However, due to the manner in which the transceivers operate, the first bit is not received at the receiver end. As a result, the first change from low to high in the transmitted byte is read as the start bit by the microprocessor or the computer. To overcome this, a particular extra byte is transmitted before the data byte. The start bit of this byte will not be received by the receiver. The next seven bits have to be low and the last bit high. In RS-232 communication protocol this corresponds to 7 high bits ("0") and one low bit ("1"). In binary this is '00000001'. Since in RS-232 communication, the data byte is sent backwards then this would correspond to '10000000' or ASCII character 128. Figure 8 shows the waveform diagram:

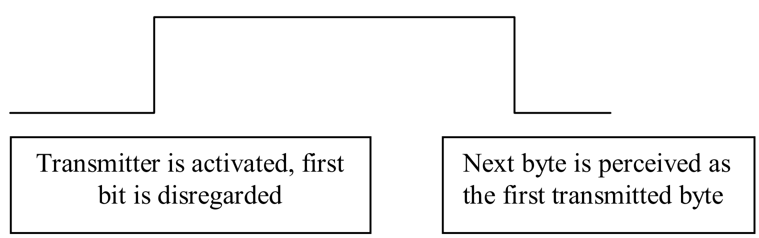


Figure 8. RS-232 Communication protocol waveform diagram

Also to ensure that the received data is a valid transmission and not RF noise, an extra security bit is transmitted before the actual byte. To ensure that the probability of replicating such a byte is low, a byte with the maximum number of transitions from high to low is used. This byte is '01010101' or character number 170.

The next byte is the valid transmission byte. Table 2 lists the valid transmission bytes.

Table2. Valid transmission bytes

Byte	Instruction
‘f’	Forward
‘b’	Backward
‘l’	Left
‘r’	Right
‘s’	Stop

This protocol is implemented by assigning the 2 states in the receive mode. The system remains in state one during which none of the received data is considered to be valid. State two is activated when the security byte is received. Any data received immediately after the security byte is considered to be valid data. The code implementation has the same form in the microprocessor in C, and in the computer in Visual Basic. The particulars of the code are discussed in some sections below.

All the specifications for the communication section of the design were achieved to satisfaction, including range, security and transmission rate.

5.0 SENSOR MODULES

The sensor modules mounted on the WatchDog serve several purposes. Infrared distance sensors and shaft encoders aid the navigation of the robot. The temperature sensor provides environmental conditions of the patrolling area. And most importantly, the human detection module checks for the presence of intruders.

5.1 HUMAN DETECTION MODULE

The human detection module is responsible for detecting the presence of intruders. Through some research, it was found that the two types of sensors that are most widely used for human detection are Pyroelectric* infrared (PIR) sensors and ultrasonic sensors. After much contemplation, it was decided that for our application, it would be most efficient and economical to use pyroelectric infrared sensors as opposed to its expensive ultrasonic counterpart. A pyroelectric sensor (RE200B) and a multi-element Fresnel lens were purchased from NICERA NIPPON CERAMIC CO. LTD.

5.1.1 Theory of Operation

The human detector module is mainly composed of an infrared pyroelectric sensor and a Fresnel lens. The pyroelectric sensor detects the presence of heat sources in form of infrared radiation, such as those emitted from humans and animals. A Fresnel lens is used in conjunction with this sensor to concentrate and maximize the IR detection.

The sensing elements of the PIR sensor are made of an infrared radiation sensitive crystalline material. A change in surface electric charge of the crystalline material is generated when radiation strikes it. This change is measured by a FET device contained in the sensor. The RE200B sensor is composed of two sensing elements internally connected in a voltage-bucking configuration. The advantage of this configuration is the immunity to environment changes. Ambient temperature changes, sunlight and vibration falsely trigger both sensing elements. In this arrangement, the simultaneous activations of both elements will cancel each other out. However, when an infrared emitting body moves horizontally across the field of view of the sensor, the two sensing elements are exposed one after the other. So, a positive or negative output signal waveform is produced, depending on which sensing element was first triggered.

These sensing elements are sensitive to a wide range of radiation, so the sensor is built with a rectangular IR filter window to limit the incoming radiation for human body detection. This filter captures IR transmission ranging from 8 to 14um. Note that the infrared radiation generated by a human body is strongest at a wavelength of 9.4um. As shown in Figure 9,

according to the manufacturer's specifications, this sensor has a field of view of 10^o by 95^o. So, in order to maximize the detection range of the pyroelectric sensor, a multi-element Fresnel lens is used to focus IR transmissions to the pyroelectric sensor. The experimental detection range obtained with the Fresnel lens mounted on top of the PIR sensor will be discussed later.

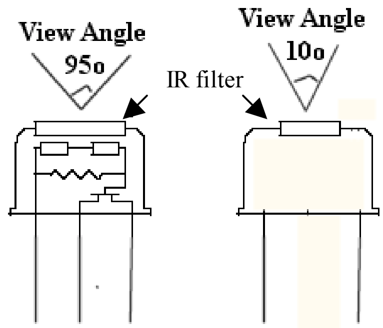


Figure 9. Detection Angle of the RE200B

5.1.2 Circuit Description

In designing a circuit for the PIR sensor, several important specifications were taken into consideration. The RE200B sensor requires a 5Vdc supply voltage. Its output has a very low frequency ranging from 0.1 to 10Hz and bandwidth. Upon the detection of a human body, the RE200B outputs an extremely low signal of about $25mV_{p-p}$. Also, as seen in the above figure, an internal transistor is found in the PIR sensor. This internal low-noise amplifier amplifies the weak differential output from the two sensing elements. Consequently, the wire connecting this signal to the pre-amplifier can easily pick up random noise, which can falsely trigger the sensor.

Also, our application requires the output of the RE200B to be interfaced with the microprocessor. So, the circuit must be designed to amplify and condition the output of the sensor before feeding it to a microprocessor. In addition, to eliminate the use of an A/D converter, the resulting signal from this circuitry should be digital, so that the output is either a high or low voltage depending on whether an intruder is detector or not. From testing, it was found that the microprocessor recognized any signal above 2.7Vdc to be high.

Numerous circuit designs were considered and tested. The first approach was to use a simple amplifying circuit, which amplifies the $25mV_{p-p}$ to at

least 2.7V. First, a non-inverting amplifier circuit, as shown in Figure 10, was tested.

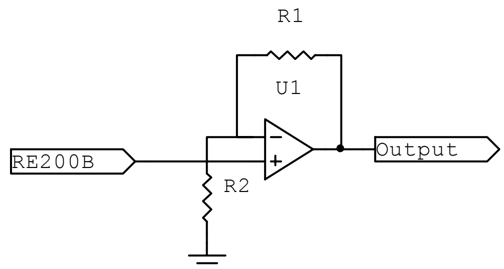


Figure10. Non-inverting amplifier

Theoretically, the output voltage of this circuit should be Vin*[(R_1+R_2)/R_2]. So, large R_1 resistors and small R_2 resistors were used. Since the input voltage is extremely small relative to the desired output voltage, preliminary testing shows that two cascaded amplification stages might be required.

In attempt to eliminate the need of two amplification stages, the use of a differential amplifier was attempted. The following figure shows the differential amplifier that was tested:

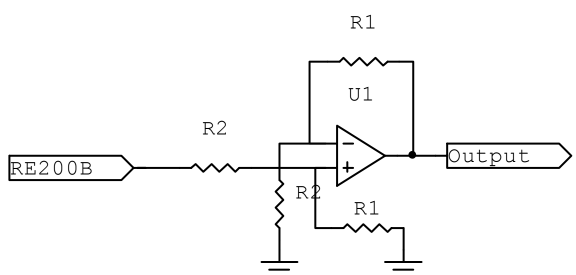


Figure11. Differential amplifier

In theory, the output of this amplifier should be -(R_1/R_2)*(0-Vin). However, this circuit didn't perform as expected.

After several other attempts, it was concluded that these simple circuits didn't provide enough sensitivity for our application. It was also noted that signal filtering should be implemented. Noise frequently activated the sensor. So, the complex circuit configuration, which was suggested by the manufacturer of the RE200B, was designed and implemented. The following figure depicts the basic configuration of the suggested human detector module:

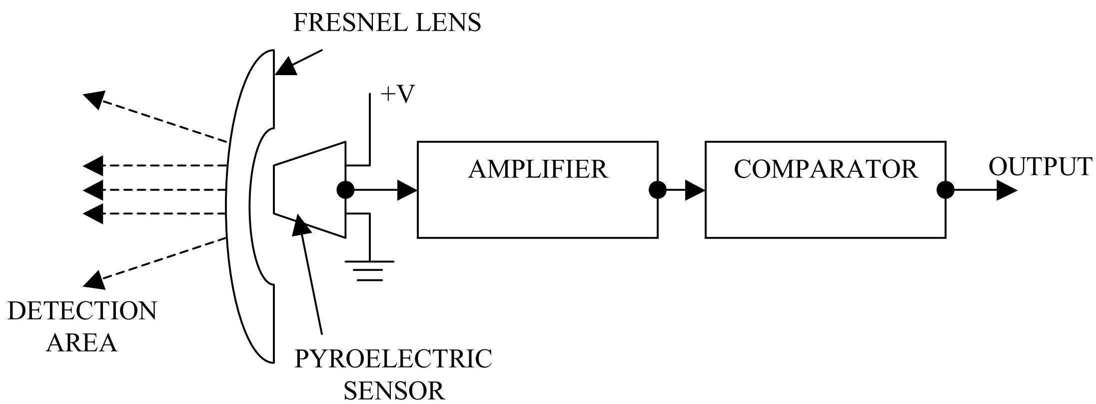


Figure12. Human Detector Module

A circuit that uses a quad operational amplifier for a two-stage amplifier and a window comparator were designed. Power of this circuit is supplied through 5V, which comes from a 78L05 regulator circuit, which is illustrated in Figure 13.

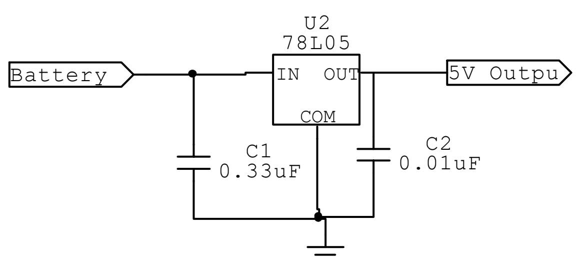


Figure13. Regulator Circuit

This 5V output from the regulator is further filtered through R_1 and C_1 before feeding it to pin 1 of the pyroelectric sensor. As previously mentioned, this sensor is extremely sensitive; slight instability in the power supply of the sensor might lead to false triggering. Each amplifier stage has a gain of 100, so the output from the pyroelectric sensor is amplified by 10,000 times. The output of the RE200B is fed to the non-inverting input of the first amplifying stage. This input is of high impedance, so the sensor is not loaded. Additional filtering was added to the amplification stages. R_4 and C_3 form a lowpass filter and feedback network. Another filter/bias network formed by R_3 and C_2 is connected to the inverting input of this amplifier. This output is then fed to $C_{4,}$ which forms a high pass filter with R_5, which blocks the flow of DC. A bandpass filter results from this configuration. The bandwidth limits only signals above DC and below 10Hz to be amplified. This filtering of signals

outside the response time of the sensor eliminates the amplification of undesired noise. Then, this filtered signal is fed to the inverting input a second amplifying stage, which is formed by a feedback network C_5 and R_8. The other input of this amplifier is connected to a resistor/diode divider network formed by R_6, R_7, D_1 and D_2. As a result, this input is biased with 2.5V. The output of this stage is connected to a comparator window. The comparator module is configured to respond to both positive and negative transitions from the sensor output. When no human is detected, its input is 2.5V. The comparator is configured to generate a positive transition when a signal of 200mV above and below 2.5V is at its input. Thus, no output is generated if small voltage transitions occur. The purpose of using a comparator is to further eliminate the false triggering of the sensor from noise. The comparator outputs are logically "OR"ed together through diodes D_3 and D_4 and are fed to the microprocessor. A positive transition is produced at the output when a human body is detected. Otherwise, the output remains at 0V. The final circuit is shown in the following figure.

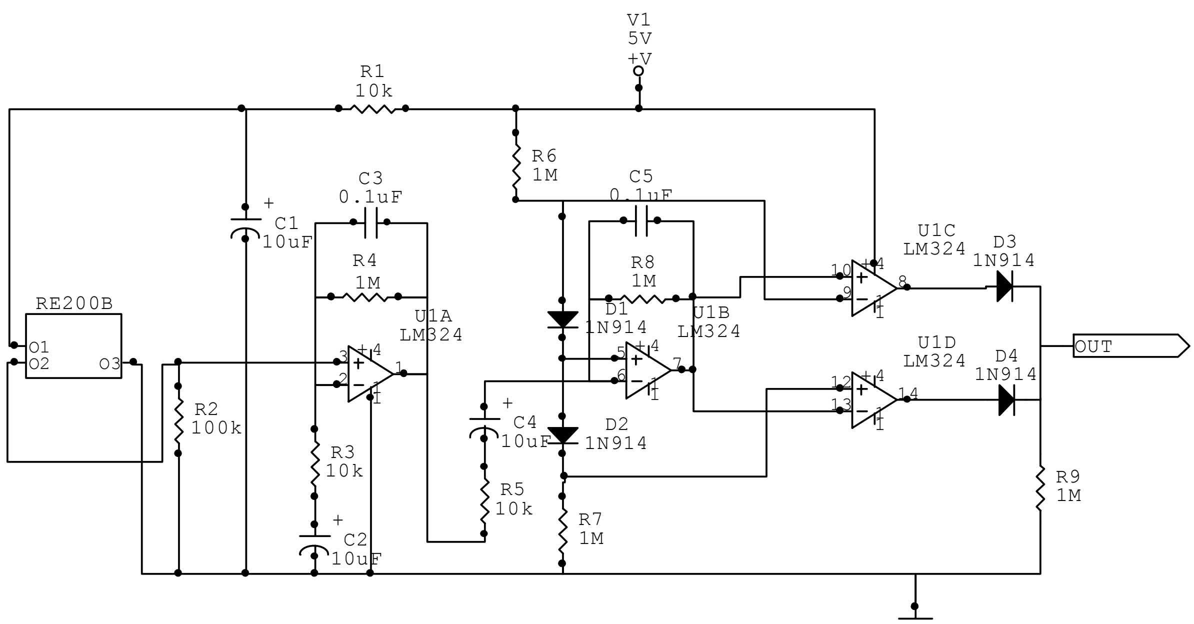


Figure14. Human Detection Circuit

An output of 4V is produced when a human is detected. The output of the comparators is not a full 5V, because of the diode drops. However, this voltage level is high enough for the microprocessor to interpret it as a high level.

It was also decided to implement direction sensing, so that the WatchDog knows the direction toward which the intruder is moving. The circuit below shows the pyroelectric sensor in a direction sensing circuit. This circuit is quite similar to the non-directional detection circuit. Because great difficulty was encountered in adjusting the sensitivity and detection range of the output in the first circuit, a 1MΩ potentiometer was used at the second amplification stage to adjust the detection range. In comparison to the previous circuit, this circuit has a slightly longer delay in its response time, and the outputs are not as stable.

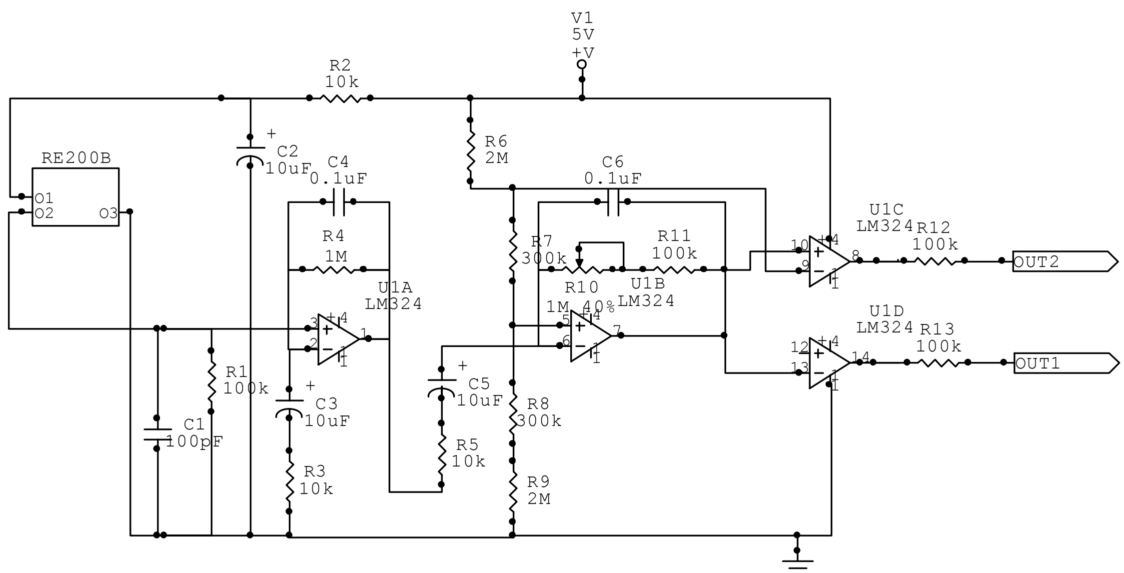


Figure15. Human Direction Detection Circuit

5.1.3 Human Detection Module Assembly

It is no use to have a good circuitry if the input signals of the sensor are dispersed. As previously mentioned, a Fresnel lens must be used to concentrate the IR detection. A wide optical zone is desirable for our purposes, so a multi-element Fresnel lens was chosen. This lens is designed for commercial motion detection systems, which are usually mounted on wall corners at 2 meters from the ground. So this lens primarily focuses IR detection below the detection module. Since our robot is always on the ground, our application requires IR detection above the robot to be focused. This can be achieved by inverting the Fresnel lens from its conventional direction. In addition, to achieve maximal detection, this lens requires a focal length of 0.9inch from the surface of the sensor. So, the rectangular lens should be curved around the sensor this focal length distance.

PCBs were designed for the above two human detection circuit. It was suggested by the manufacturer that the sensor should be mounted on the back of the board, so that all the other circuit components are on the other side.

Minor air fluctuation caused the detection circuit to be falsely triggered. Since this module is to be placed on a mobile robot, the slightest vibration, which would cause air fluctuation at the surface of the sensor, leads to a false output. Thus, the module must be shielded in a closed box. A metal box was made to contain both the PCB and Fresnel lens. It metal could not be bent circularly with the desired focal length of the lens, so the lens was simply placed flat on the box. The detection module worked perfectly, however this metal box was relatively heavy for the robot to carry. So it was replaced with a cardboard box. Cardboard also has the advantage of being more bendable, so it was possible to mount the lens with the desired the focal radius. The cardboard detection module was tested. It was found that the narrow gaps between the edges of the cardboard surfaces allowed the passage of air, which led to false triggering of the module in a mobile environment. So the cardboard box was entirely shielded with tape. The final detection module is shown in Figure 8.

Figure16. Human Detection Module

This module was tested. By adding the multi-element lens, the original field of view of 10^o by 95^o was extended to approximately 70^o by 120^o. The detection range of the sensor was measured to be 4m.

5.1.4 HC11 Interface

Both directional and non-directional human detection circuits work perfectly. However, due to the lack of input pins on the microprocessor. The directional circuit would have required two inputs from the HC11, one for each direction (left/right). So, it was decided to use the non-directional detection circuit, which requires only one input line from the HC11. The output of the human detection is fed to the IRQ of the HC11.

5.1.5 Difficulties Encountered

Great difficulty was encountered in adjusting the signal to noise ratio of the detection circuit. Various amplification gains and comparator reference levels were tried. Having a larger comparator window allowed amplified noise to produce false outputs, whereas having a smaller comparator window and amplification gain didn't provide a signal with a sufficiently high level, so the output remained indifferent upon the detection of a human. The component values in the final circuits shown in this report were found to be the most appropriate.

It was also hard to test the sensor from a prototype circuit that was on either the breadboard or soldered on a perfboard. The circuit was unstable due to noise issues. It was also hard to align the center of the lens with the center of the sensor with the required focal length. However, once the components were soldered onto the PCB and shielded in a box. These problems were eliminated.

5.1.6 Design Evaluation

The finalized non-directional human detection circuit has both strengths and limitations.

5.1.6.1 Advantages

The goal of having a digitized output from the detection circuitry for easy interface with the microprocessor was achieved. In addition, the noise issues were successfully eliminated. The human detection module provides accurate outputs for a wide detection range.

5.1.6.2 Disadvantages

A significant limitation of the human detection circuit is the warm-up period of approximately 15 seconds when it is first powered.

The capacitors of the amplifier and the RE200B sensor require time to charge up and stabilize to their normal operating conditions before the correct operation of the circuit. Thus, the circuitry will not respond to human detection until about 15 seconds after power is applied.

It is important to note that the RE200B sensor output is triggered if only one of its two sensing elements is exposed at a time. This occurs when a human moves into or out the field of view of the WatchDog. So, another drawback is the inability of the sensor to detect the presence of a human standing still perfectly in front of the sensor. This is because both sensing elements will be triggered at the same time, therefore canceling each other's signal. However, for our application, this should not be problematic, because the robot is mobile. Whether an intruder is moving or standing still, as the WatchDog moves, the intruder will move be in and out of the field of view of the detection module, thus creating a change between the two sensing elements.

Another drawback of this sensor is its need of a constant power supply. Due to the amplification of the detection circuit, it is easily triggered with the slightest line voltage fluctuation. As previously mentioned, a filter was added between the input of the sensor and the voltage regulator. This filtering works appropriately. However, if several other circuitries share the same battery supply with the human detection circuit, larger line voltage fluctuations may occur, and won't be filtered. So the human detection circuit may require a separate battery that might not be shared with other circuitry.

5.2 DISTANCE SENSORS

Distance measuring sensors are used to detect the presence obstacles in the navigation path of the robot. Distance sensing is most commonly done using infrared proximity sensors. Sharp provides a wide array of accurate and compact short detection range sensors. The GP2D02 from Sharp was found to be the most appropriate distance for our application. According to manufacturer's specification, it has a detection range of 10-80cm. This sensor is impervious to the color and reflectivity of objects, and is immune to ambient light. This sensor provides a serial digital output, which can be directly interfaced to the microprocessor, so no signal conditioning is required.

According to the manufacturer's specifications, the GP2D02 has a narrow field of view of 10cm. The WatchDog will use two GP2D02s. Both of these proximity sensors will be mounted at the front of the robot (one on the left and one on the right).

5.2.1 Theory of Operation

The GP2D02 consists of an infrared LED, an array of position sensitive photodiodes and optical lenses. The infrared LED emits a straight beam. If the beam hits an object, part of that infrared beam is reflected back towards the sensor. The reflected beam that passes through the receiver optics, is then focused onto the array of photodiodes. The distance of the object can then be determined from the position sensitive photodiodes. Figure17 illustrates the operation of the GP2D02.

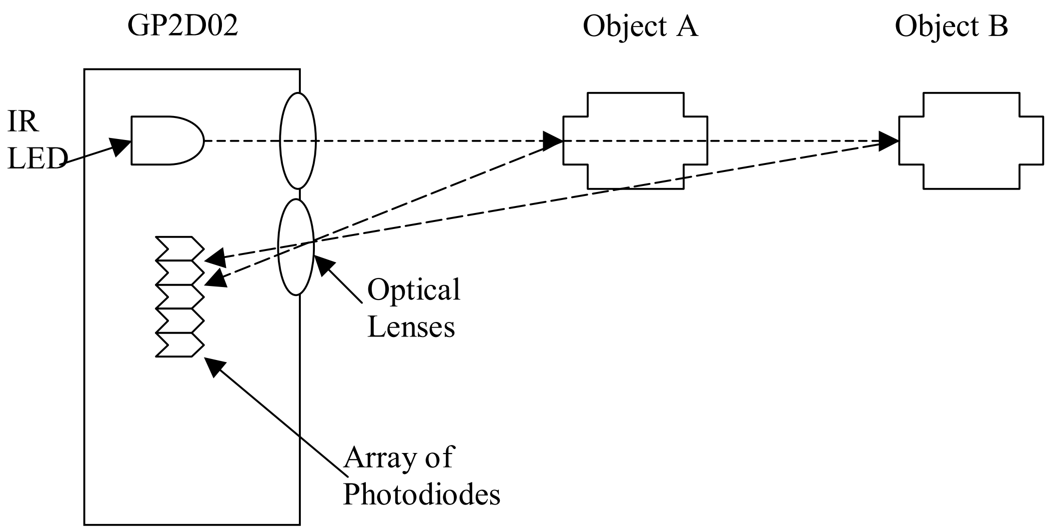


Figure17. Operation of the GP2D02

From the above figure, it can be seen that the photodiode that is hit by the reflected beam is dependent on the distance of the object. The reflected beam from object A forms a greater angle at the receiver lens than that of object B. Notice that if an object was further than object B, its reflected light would not hit the array of photodiodes. Similarly if an object is extremely close to the sensor, its reflected beam would also not hit the array of photodiodes. The array of photodiodes in GP2D02 is configured for detection ranges of above 10cm and below 80cm.

The array of photodiodes are arranged in a voltage divider configuration. When a certain position sensitive photodiode is hit by a reflected beam, a current flow is generated by that photodiode. When the currents of all the photodiodes are compared, a voltage, that is dependent on the activated photodiodes is generated. This voltage is then digitized into an 8-bit serial output.

The GP2D02 has 4 pins, which are connected in the following configuration:

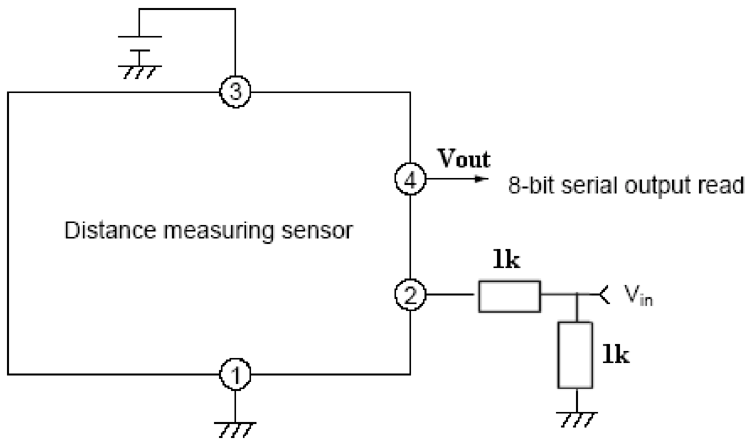


Figure 18. Connection Diagram of the GP2D02

This sensor requires a supply voltage of 5Vdc at pin3. V_{in} is an output from the microprocessor which controls the distance measurement of the sensor. A voltage divider is connected between this pin and the output from the microcontroller. This is because pin2 can only tolerate a maximum input voltage of 3V. So, the high level output (5V) of the microncontroller must be reduced through the voltage divider.

The GP2D02 must be initiated via a microprocessor to activate a distance measurement. Figure 19 is the timing chart from the manufacturer, which describes the required steps.

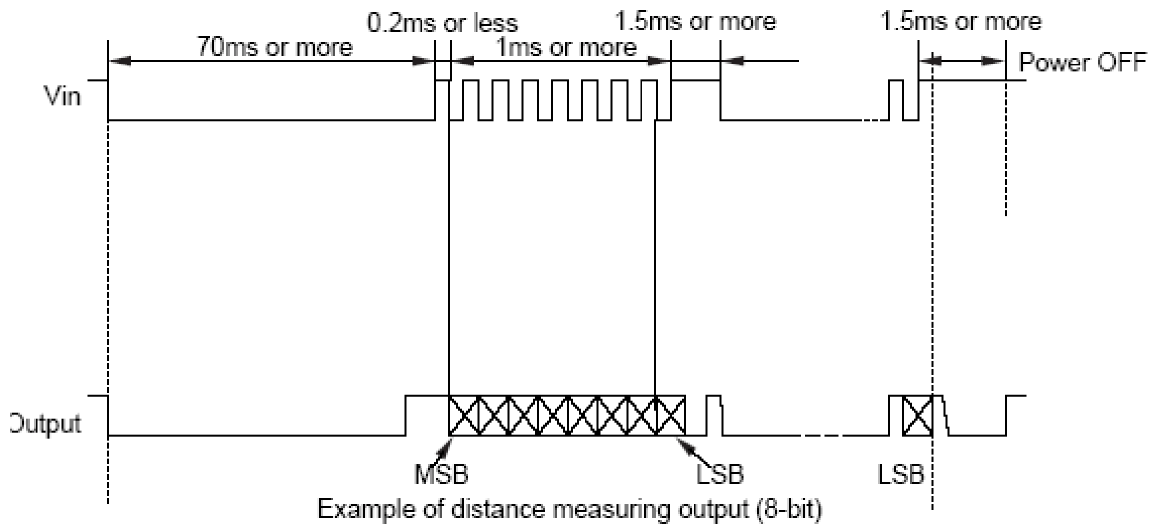


Figure 19. Distance Measurement Timing Chart

First, the output of the microcontroller (V_{in}) must be held "LOW" for approximately 70ms, until the output (V_{out}) goes "HIGH". Then, the distance measurement is initiated by generating a series of 8 pulses at V_{in}, one for each bit. Each pulse must be less than 0.2ms. After each pulse, an output bit must be read by the microprocessor at V_{out}. At the end of the 8 pulses, the sensor is reset by setting V_{in} to "HIGH" for at least 1.5ms to reset. The 8 bit output obtained at V_{out} is a decimal value.

5.2.2 Calibration

The output distance obtained from the V_{out} of the sensor is not the actual distance of the object. So, this output must be calibrated. As mentionned by the manufacturer, the different sensor units may differ from each other slightly. So, the calibration must be done seperately for each of the two sensors.

The first approach for testing the sensors was to manually do the steps described in the previous section, without the use of the HC11. V_{in} was grounded, then it was pulled out and a rectangular pulse was immediately fed from the function generator. This attempt was unsuccessful. No output was generated at the output of the sensor. Thus, a program which conforms to the steps needed to control the distance measurements was written for the HC11.

Then, experiments were conducted to investigate the relationship between the output from the sensors and the actual distance. The results obtained for the two sensors are plotted along with the manufacturer's experimental data.

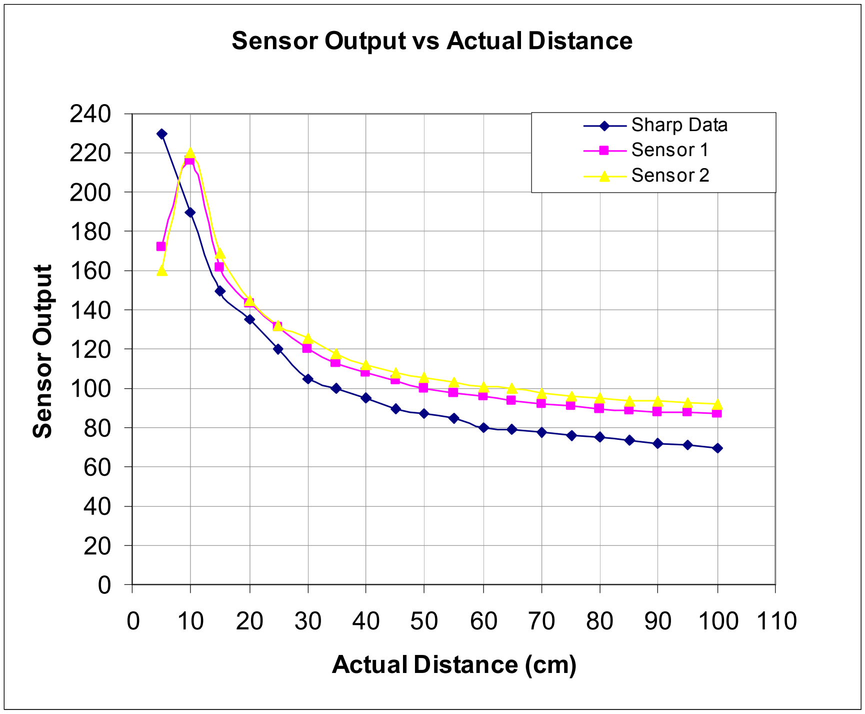


Figure 20. Experimental Data of two GP2D02s

From the above graph, it can be deduced that the relationship between the the sensor output and the actual distance is non-linear. The results obtained from both sensors are quite similar, so the calibration of both sensors may be done with the same calibration values. Note that the results from both sensors differ quite significantly from the manufacturer's claims.

An expression must be determined to calculate the actual distance from the sensor output. To facilitate calibration calculations, the curve for each sensor can be split into 4 regions:

1) Actual distance below 20 cm:

The peak in the above graph shows that the values below the actual distance of 20cm should be disregarded, because the sensor output may represent two different actual distances. Thus if the sensor output value is below 145, which means that the actual distance is less than 20cm, no calculation is done. The value is simply rejected; the sensor cannot perform correctly beyond this limitation range.

2) Actual distance between 20 and 50cm:
The actual distance for sensor output values between 145 and 100, which correspond to an actual distance between 20-50cm, can be calculated using the following equation:

Actual Distance = 1300/(*Sensor output*-80)

3) Actual distance between 50 and 80cm:
The actual distance for sensor output values between 100 and 90, which correspond to an actual distance between 50-80cm, can be calculated using the following equation:

Actual Distance = 339-2.73**Sensor output*

4) Actual distance greater than 80cm:
Sensor output below 90, which correspond to an actual distance of greater than 80cm should also be rejected.

The calculated actual distance for both sensors is plotted along with the experimental data obtained earlier to compare the accuracy of the above calculation methods.

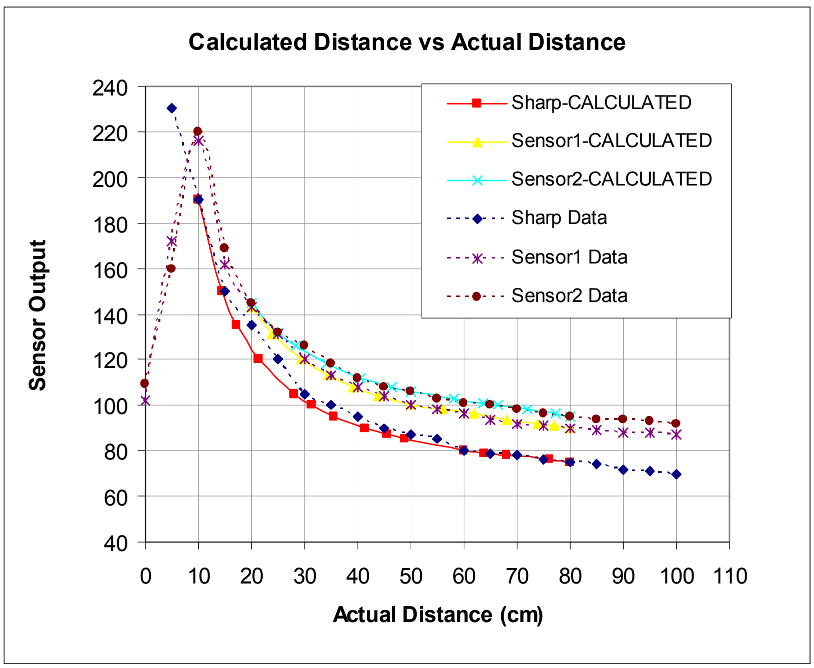


Figure 21. Calculated Distance of the GP2D02s

The above plot shows that the calculated value using the experimental data is quite close to the actual distance. So the above equations can be used for distance calibration.

5.2.3 Design Evaluation

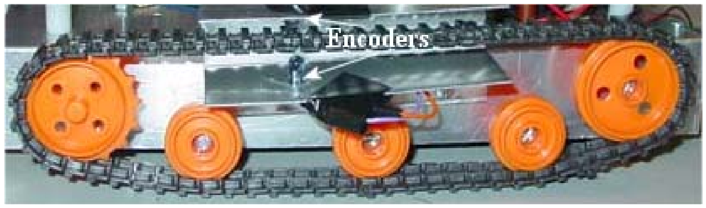
The software for the sensor was fairly simple to implement. Calibrating the sensors was time consuming, but the resulting calculated distance is quite accurate. A drawback of the GP2D02 lies in the repeatability of accurate results. The output obtained by the sensor is not perfectly repeatable. Thus, the calibrating data used in the calibration equations may differ slightly from a certain measurement, which would lead to slight errors in its calculated distance. However, precise reading of a proximity object is not required for our application. So, the non-repeatability of the sensor is not problematic. The WatchDog only needs to know if an obstacle is nearby. This will be done in the software by setting a certain threshold value. If the reading from the GP2D02 is greater than that threshold value, than an obstacle is present. Otherwise, no obstacle is present.

5.3 TREAD ENCODERS

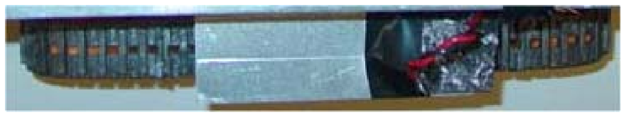
To facilitate the navigation of the WatchDog, optical encoders, composed of an infrared LED (Radio Shack Part#276-145) and an infrared sensitive phototransistor (Radio Shack Part#276-143), are mounted onto the treads of the robot.

5.3.1 Theory of Operation

The infrared LED acts as an emitter that emits infrared energy of about 880nm. The infrared sensitive phototransistor acts as a receiver. A base voltage of the phototransistor the base voltage is determined by the amount of infrared light that hits it. So, a phototransistor can be seen as a variable current source. The amount of infrared light is directly proportional to the amount of current that flows through the collector and emitter.



(a)



(b)

Figure22. Mounting of Encoders (a) Side view; (b) Top view

The above picture shows the placement of the encoders. As seen from the top view of the treads, there are holes between the ridges. When the infrared LED and the phototransistor are blocked by a ridge, no infrared light is received by the phototransistor, so the voltage output should remain 0V. On the other hand, when the encoders are not blocked by a ridge, the infrared light emitted is received by the phototransistor, which produces an analog output voltage.

5.3.2 Circuit Description

A digitized output is desired from this encoder module. The output of this circuit should be 0V when no infrared light is received and a high level, which can be detected by the HC11, when infrared light is received. A simple circuit, as shown in Figure 23, is implemented to test the encoders. The phototransistor is connected in a voltage divider configuration. The variable current traveling through the resistor causes a voltage drop. The values of R_1 and R_2 were chosen so that the output voltage obtained at the phototransistor side is maximal. The maximum output obtained was 2V. This output was fed to the controller, which failed to recognize the high level signal. So, the output of the phototransistor must be amplified.

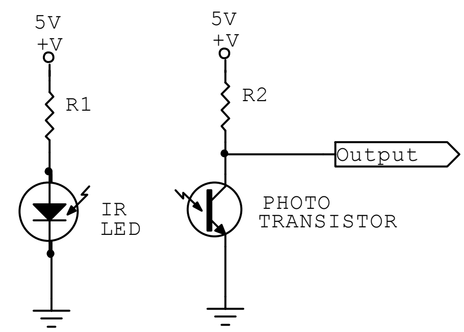


Figure23. Simple Circuit for Tread Encoders

Unsuccessful attempts were made to amplify the output from the above circuit using a non-inverting amplifier, an inverting amplifier, a Schmitt trigger and a differential amplifier. Then instead of using op-amps, transistors were used to accomplish the required amplification. A configuration consisting of two amplifying stages was implemented. First, the output from the phototransistor is amplified through a common-emitter transistor network, which has the advantage of high voltage gain from a relatively low input impedance. Then the high impedance output from this stage is fed to a common-collector transistor stage, which converts the high input impedance to a low output impedance. This amplified output is then digitized through a Schmitt Trigger.

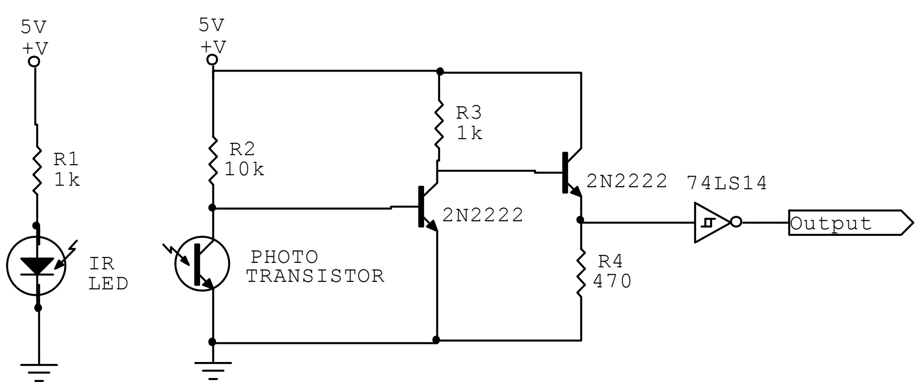


Figure24. Tread Encoders Circuit

In the case that the emitter IR LED successfully transmits an IR beam to the phototransistor, the output current flows to the base of the first amplifying stage, but no current flows to the base of the second amplifying stage. So, a high level output is generated by the Schmitt Trigger. In the case that the IR LED and the phototransistor are blocked by a ridge of the treads, no voltage signal is generated at the phototransistor output, so current flows through R_3 to the base of the second transistor. So a low level output is generated by the Schmitt Trigger.

5.3.3 HC11 Interface

The pulses generated by this encoder circuit are fed into the input capture pins of the microprocessor. Each time a positive transition occurs, the HC11 is configured to generate an interrupt, which increments the count number of the pulses. The software successfully counted the number of pulses generated by both encoder circuits. However, interrupt handling issues become problematic when this code is combined with the RF software, which also makes the use of interrupts.

5.3.4 Difficulties Encountered

Great difficulties were encountered in finding suitable resistor values for the IR LED and phototransistor circuits. The mounting of the encoders also presented a challenge. Because the chassis of the robot is made out of metal, whenever both leads of either the LED or phototransitor touched the chassis, the shorting would blow the transistors of the circuit. Another mounting problem can be attributed to the small hole sizes of the treads. The encoders were not able to capture all the pulses when the treads are moving. This issue was resolved by aligning the LED and the phototransistor with an angle to one another. It was observed that all the pulses were properly measured when both components were placed at an angle of each other.

5.4 TEMPERATURE SENSOR

It was decided not to have a fire detection module due to its cost and a shipping and handling time of 6 weeks from the manufacturer. Instead, a temperature sensor is implemented. An Econo 1-Wire Digital Thermometer, DS1822, was obtained from Dallas Semiconductors.

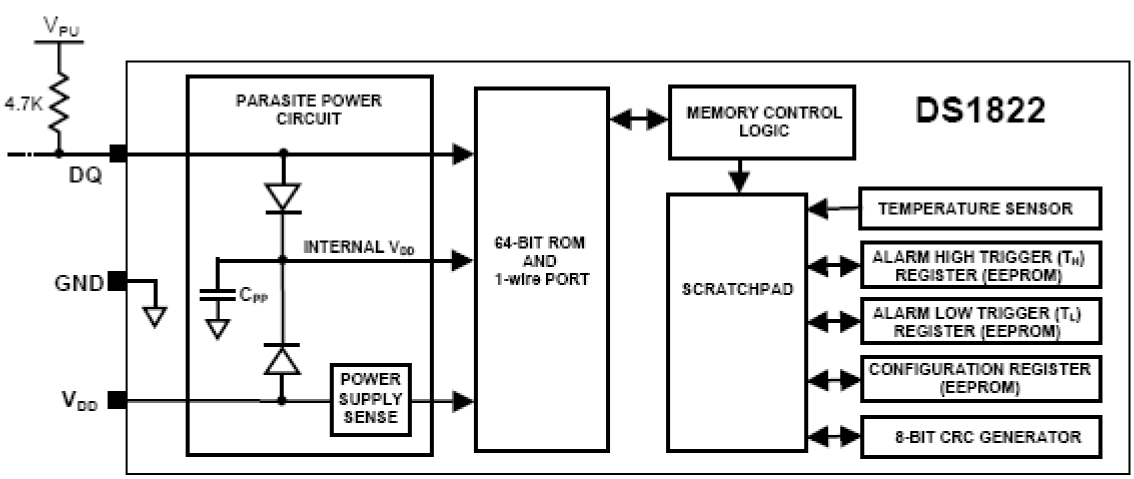


Figure25. Block Diagram of DS1822

The DS1822 is a 1-Wire temperature sensor, which requires only one bi-directional pin from the HC11 for communication. No external circuitry is required; it can be directly powered from a supply voltage of 5.0V. It can measure temperatures from -55°C to +125°C. The accuracy of the measurement should be within 2°C for temperatures ranging between -10°C to +85°C. This unique thermometer is user-programmable. The sensor has a 64-bit serial code stored in a built in ROM, which allows user to select the thermometer resolution.

5.4.1 Theory of Operation

The temperature measurement of the DS1822 must be controlled via the DQ pin. First, the sensor must be initialized, through a reset pulse from the master followed by a presence pulse from the sensor. This is shown in Figure 26, which is a timing chart provided by the manufacturer.

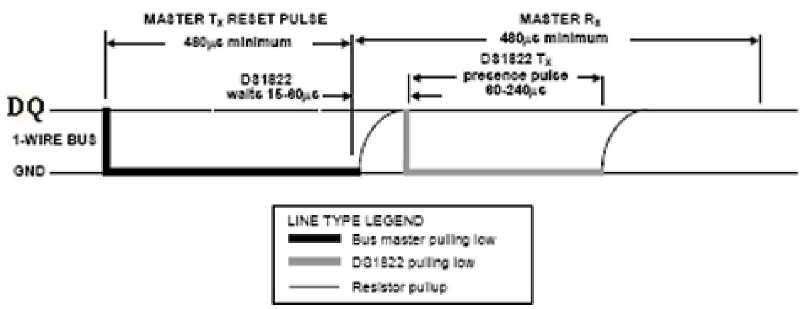


Figure26. Timing Chart for Initialization of DS1822

The reset pulse is generated by pulling the 1-wire bus (DQ) low for at least 480us. Then, the transmit mode of the bus is switched into the receiving mode. In response to the reset pulse, the DS1822 should wait for about 15-60us and respond by transmitting back a presence pulse, which indicates to the master that the sensor, on the bus, is ready to operate.

The temperature conversion is controlled by writing and reading data to the DS1822 via the 1-wire bus. One bit of data is transmitted over the 1-Wire bus per time slot (see Figure27).

The bus only receives data when the master issues a write time slot. The bus may be written as a logic 1 or a logic 0. Writing a bit requires at least 60us with a minimum recovery time of 1us between each bit writing. A logic 1 is written by pulling the 1-Wire bus low, and releasing the 1-Wire bus within 15us. When the bus is released, the 5k pull-up resistor pulls the bus high. A logic 0 is written by pulling the 1-Wire bus low, and holding the bus low for at least 60us. The 1-Wire bus is sampled whenever the master initiates a write time slot. If the bus is high during the sampling, a 1 is written to the DS1822. If the bus is low, a 0 is written to the DS1822.

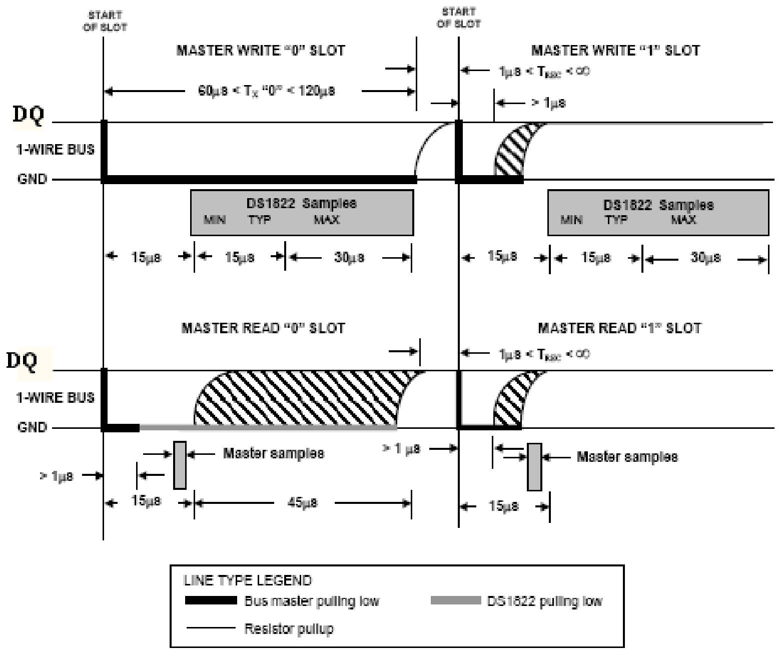


Figure27. Timing Chart for Reading/Writing to the DS1822

The DS1822 transmits data when the master is set to read time slots. Data is read after the initialization of a temperature conversion. Reading a bit requires a minimum of 60us with a minimum of a 1us recovery time between readings. A bit reading is initiated by pulling the 1-Wire bus low for a minimum of 1us and then releasing it. A logic 1 is transmitted by leaving the bus high, whereas a logic 0 is transmitted by pulling the bus low. Following the reading, the bus is released and pulled back to its high through the pull-up resistor.

5.4.2 HC11 Interface

A program which conforms to the above steps was written for the HC11. First, the DS1822 was reset and a temperature conversion was initiated by writing to the master bus. Then the 9-bit serial temperature is read. This serial output from the sensor is the decimal value of the actual temperature, so no further calibration is required.

5.4.3 Evaluation of Design

The DS1822 was easily interfaced with the HC11. The program written successfully initiated and converted the temperature into a 9-bit serial output. The temperature sensor was not tested for various temperatures, because that would require testing this module at various environments. However, it seems to perform correctly at room temperature.

6.0 THE CAMERA

We used the Xcam2 color camera from X10 Technology Inc. for the WatchDog. Figure 28 shows a front view of the camera.

Figure28. Front view of the Xcam2 color camera

The specifications of Xcam2 [1] are the following:

- Imager CMOS Sensor
- Format 1/3"
- Array Size NTSC 510 × 492 pixels
- Resolution 310 TV Lines
- Scanning 2:1 Interlace
- Auto Shutter 1/60 to 1/15,000 second.
- Minimum Illumination f1.8
- Operating Temperature -10°C to 40°C (14°F to 104°F)
- Humidity Limits 0 - 95%
- 60 Degree Range of View
- Transmission Frequency 2.4 GHz

The Xcam2 colour camera integrates with a 2.4 GHz transmitter that works in analog mode. Such high carrier frequency provides a relatively wide range of operation (approximately 100 ft) and less interference from the environment. A video receiver is also included in the camera package. The receiver receives the modulated video signal and demodulates it. The output from the receiver is standard video out. The camera and the integrated transmitter function on 12V DC at 80mA. A minimum voltage of 10.5V at 64mA has to be maintained in order to ensure proper operation of the camera. The camera by default requires 0.96W of constant power supply, which is quite enormous when compared to other electronic components in the robot.

To communicate with the monitoring PC, a separate video capture device is required. Digital Video Creator 50 (DVC 50) made by Dazzle Multimedia Inc. is utilized to digitize the video signals and then to transfer the digitized signals to the PC. This device

collects signal from the video receiver through standard video cable and sends 16-bit digital signal to the monitoring PC via USB* 1.1 interface. This interface provides data transfer and supplies the power that drives the device. The video receiver runs on 9V DC at 100mA. To power this receiver, the power supply in the camera package converts 110V AC to 9V DC and outputs 100mA. The video receiver is not mounted on the WatchDog and therefore its power consumption is not a critical issue.

7.0 SOFTWARE DESIGN

The monitoring PC has an interface that operates along with the robot. The interface mainly contains two components. One is the software that captures and stores data from the camera through DVC 50. The other is the software that uses RF devices to communicate with the robot through bi-directional communication. The RF module at the PC end connects to the PC via a standard RS-232 (EIA232) serial port*.

The camera software was implemented first. The communication software was designed later. Both were tested independently, and then combined together. During the planning stages of the software development, we decided that we wanted to be able to easily incorporate the components together therefore; we developed them under the Microsoft Visual Basic 6.0 platform.

7.1 ADVANTAGES OF IMPLEMENTING IN VISUAL BASIC

The advantages of VB are as follows:

1. VB provides a graphical programming interface for Windows applications. The amount of the source code in VB is minimal when compared to other Windows programming languages.

2. The video capture/control function calls offered by VB do not require programmers to deal with any low level PC interfacing issues. That is, no matter what interface the video capture device uses (Serial Port, USB, IEEE 1394*, or Bluetooth*) programmers can always have access to any of these devices at a high programming language level. The operating system takes care of the low level issues.

3. The communication software connects to the RF PC end module through standard RS-232 interface. Communication with RS-232 can be easily implemented by VB 6.0's ActiveX Control component MSComm32.OCX.

4. The complexity of the software requires many Win32 API* function calls, such as Sleep( ), GetKeyboardState( ), SetKeyboardState( ), and SendMessage( ). Because Visual Basic is a development tool implemented by Microsoft, it has great flexibility of accessing standard Win32 API calls from within the code. Also, the speed of such function calls from VB is fairly high as examined through the development of our software.

7.2 SOFTWARE

In writing the software, some sample codes from E. J. Bantz [2] were consulted. Many features and functions were added to the sample code and modifications were made to finalize the software.

7.2.1 Camera Software Design Process

The software design process can be summarized in the following steps:

Step 1. Creating the window to capture video
Step 2. Linking the form window to the video driver
Step 3. Passing strings to SendMessage( )
Step 4. Passing structures to SendMessage( )
Step 5. Processing/storing images from the video stream

7.2.1.1 Step 1. Creating the Window to Capture Video

The video capturing process begins with the capCreateCaptureWindowA( ) function. When the function is called a handle to a new window is to be returned. The returned handle is a 32-bit number used to refer to an object, in this case a window. This window is a child* and is visible* once created. The following code demonstrates the usage of the capCreateCaptureWindowA( ) function:

```
lwndC = capCreateCaptureWindowA("My Capture
Window", WS_CHILD Or WS_VISIBLE, 0, 0, 160, 120,
Me.hwnd, 0)
```

All future commands are issued by sending a Windows message to the new window. A Windows message is sent through the SendMessage( ) function that is built as a part of the Win32 API. When passing the function, a handle, which is a 32-bit parameter, is to be returned.

7.2.1.2 Step 2. Linking the Form Window to the Video Driver

When the capture window is created, it is connected to the video driver by the use of the WM_CAP_DRIVER_CONNECT message, which binds the window to the first video driver (with index 0) that it can find. Also, the final user will have the opportunity to select a desired video driver from a list of drivers in a separate window. The video driver is then sent to any window that exists, but only the window created by capCreateCaptureWindow will have access to it. This retrieving process protects the program from the access of other active

windows. The usage of the WM_CAP_DRIVER_CONNECT message and the SendMessage constant is demonstrated below:

```
SendMessage lwnd, WM_CAP_DRIVER_CONNECT, 0, 0
```

Aside from defining all the possible message constants, VBAVICAP.BAS provides many useful functions that make sending windows messages much easier. For instance, making the following function call is exactly the same as the one above:

```
capDriverConnect lwnd, 0
```

Therefore, these functions are adopted instead of SendMessage( ) to avoid some confusing cases, in which the parameters are passed by SendMessage( ).

7.2.1.3 Step 3. Passing Strings to SendMessage( )

The SendMessage API is always passed in order to return four different parameters: a handle to a window (hwnd), a message that sends to that window (wMsg), a short parameter (wParam), and a long parameter (lParam). The declaration yields the following code:

```
Declare Function SendMessage Lib "user32" Alias
"SendMessageA" (ByVal hwnd As Long, ByVal wMsg As
Long, ByVal wParam As Integer, ByVal lParam As
Long) As Long
```

To facilitate passing a string to this function, it is a decent approach to redefine the lParam as a String instead of as a Long.
However, a Long parameter must be passed to a DLL*. When passing a String by Value to any external functions, VB actually sends a pointer to the String instead. The pointer is a 32-bit number, which is also a Long. Hence, the proper variable type is passed to the function as expected. Note that the declaration of SendMessage( ) must be fixed from any point of the code and must be in the Option Explicit section because some calls require lParam to be a Long.

7.2.1.4 Step 4. Passing Structures to SendMessage( )

In VB, structures cannot be directly passed to the functions. Instead, passing structures can be achieved by the use of the function VARPTR( ). This build-in function returns the location in memory, which is a pointer where the passed variable is stored. In

particular, the variable is a User Defined type. The following declaration can retrieve the capture parameters:

```
Dim CAP_PARAMS As CAPTUREPARMS
capCaptureGetSetup lwndC, VarPtr(CAP_PARAMS),
Len(CAP_PARAMS)
```

7.2.1.5 Step 5. Processing/Storing Images from the Video Stream

VB provides some powerful callback functions in charge of video/image processing. For instance, capSetCallbackonFrame( ) is used to process frames during preview, whereas capSetCallbackOnVideoStream( ) can process frames during capture. The declaration can look like the following:

```
capSetCallbackOnFrame lwndC, AddressOf
MyFrameCallback
```

where MyFrameCallback( ) is a function call located in a module, not in a form*. This function is called every time a new video frame is received from the video driver.

7.2.2 Conflict with Default Driver

The driver source of DVC 50 contains several files that include:

NUVAUDIO.INF
NUVAUDIO.SYS
NUVISI02.INF
NUVISION.AX
NUVISION.CAT
NUVISION.DS
NUVISION.INF
NUVISION.SYS
NUVTWAIN.DLL
NUVYUV.DLL

Among these files, NUVYUV.DLL, which is developed by Dazzle Multimedia, is in charge of the DVC50 default video acceleration mode YUV. When installing the driver to Windows, Windows adds four more DLL files developed by Microsoft. These files correspond to four other video acceleration modes that are YUV2, RGB24, RGB555, and I420. The video card in the monitoring PC, on which the program is tested and run, automatically adopts the default mode YUV. However, this mode causes a

fatal conflict with the operating system, in particular Windows 2000, every time the program runs for approximately 1 minute.

While debugging the file NUVYUV.DLL, it was discovered that one stack pointer points to an address location that causes overflow of memory and then crashes the system. This means that the driver file is not implemented well and therefore the bug is exceptionally difficult to be fixed since no technical design details or supports of DVC 50 are available.

We decided to resolve this problem by simply deleting the file NUVYUV.DLL under the directory WinNT\System32\. After deletion, the video card automatically switches to another acceleration mode. It switched to YUV2 on the lab computer. The new mode works fine with the DVC 50.

7.2.3 Resolution and Picture Quality

When calling the Format_Click( ) or mnuFormat( ) function, the resolution of the video frames can be modified. The modification process is a low level process controlled by the operating system. The different resolutions DVC50 supports are 160x120, 176x144, 240x180, 320x240, and 352x288. The default is set to 320x240.

Changing the resolution significantly influences the picture quality. In general, the higher the resolution, the better the picture quality. However, the distinction between the 352x288 and 320x240 resolutions is not possible with the human eye. This is due to the hardware of the Xcam2 camera, which supports a maximum of 310 horizontal lines. Increasing the resolution simply results in an arbitrary insertion of video lines by the operating system. Therefore, the quality cannot be improved. The higher than default resolution also introduces more delay and uses more CPU resources because this resolution is not standard. Therefore, 352x288 is not recommended, whereas 320x240 is appropriate for the application.

7.3 COMMUNICATION SOFTWARE

The main function of the communication software is to provide two-way communication with the RF terminal, which allows the monitoring PC to exchange information with the WatchDog. The microcontroller sends requests to the PC and the PC sends commands to the microcontroller, both through the RF wireless channel.

The communication port that the PC uses to connect the RF terminal is Serial Port (COM1) by Standard RS-232 (also known as EIA232). VB 6.0 provides an ActiveX control named MSComm32.OCX to work with the COM ports. The functions built in MSComm32.OCX allow programmers to easily control, transmit and receive data from the port. However, the current version of Visual Basic, Visual Basic.Net, no longer offers the MSComm32.OCX for unknown

reasons.[3] Accessing the COM ports under VB.Net therefore becomes much more complicated. This is another reason we chose the programming language to be VB6.0.

7.3.1 Interface with Hardware

The RS-232 supports several operation modes, including hardware handshaking, software handshaking, and no handshaking. From another perspective, the receiving modes can be classified into two groups, polling or interrupt driven.[4]

7.3.1.1 Operation Modes

In the standard handshaking mode using the RS-232 DB9S interface, DTR (pin 4), DSR (pin 6), RTS (pin 7), and CTS (pin 8) pins are involved. When a node wishes to transmit data it asserts the RTS line active high. It then monitors the CTS line until it goes active high. If the CTS line at the transmitter stays inactive, it means that the receiver is busy and not available. When the receiver reads from its buffer the RTS line becomes active high, indicating to the transmitter that it is now ready to receive a character. Receiving data is similar to the transmission of data, but the lines DSR and DTR are used instead of RTS and CTS.

Two ASCII* characters, start and stop communications during the software handshaking mode. These are X-ON (^s, or Chr(11)) and X-OFF (^q, or Chr(13)). When the transmitter receives an X-OFF character it ceases communication until an X-ON character is sent. Software handshaking is generally used when both transmitter and receiver can process data at a relatively high speed. It also requires both transmitter and receiver to have large buffer capacities.
When no handshaking is involved, the software simply assumes the transmitter and receiver can process data without any loss, and no hardware pins or special software characters are involved. This is suitable in times when data accuracy is not that important.

The RF terminals in our project could not provide a high data transfer rate however they required accurate data transfer. As a consequence, a simplified hardware handshaking method is adopted. That is, only DTR pin is utilized. When PC intends to send data, it enables DTR for 1 second. Within this period of time, actual data are transmitted. The RF module is only activated to transmit or receive data when DTR goes high. This method is tested many times and no data is lost through transmission.

7.3.1.2 Receiving Modes

Polling is a process where the program checks if there is any incoming data through the COM port at a period in time, which can be set in the program. If data is received, an event is triggered so the program processes the data. Otherwise, the program will work on other processes until the next time period.

Interrupt driven mode works in the reverse manner. The program will check and process the incoming data only if an interrupt is activated. Otherwise, the program does not check the COM port.

The interrupt driven mode is efficient on processor time as it allows the processor to run a program without having to poll the serial port. It allows almost instant access to the processor and stops slow devices, such as the serial port, from reducing the performance of the processor. However, the interrupt driven process requires several Win32 API calls that will slow down the program. Also, the low amount of data exchange does not suffice this rather complicated algorithm. Thus, only one byte is transmitted/received at a time. Therefore, the interrupt driven mode was not beneficial and we utilized the polling mode.

7.3.1.3 RS-232 Communication Settings

As mentioned before, the data exchanged per transmission/reception is three characters or bytes. Thus, our serial port setting is: Serial Port COM1, 8 information bits, no parity bit, 1 stop bit, and the baud rate is equal to 300 bits/second. The VB expression is

```
MSComm1.CommPort = 1
MSComm1.Settings = "300, N, 8, 1"
MSComm1.InputLen = 0
MSComm1.PortOpen = True
MSComm1.Handshaking = comNone
MSComm1.DTREnable = False
MSComm1.RTSEnable = False
```

When loading the form, handshaking is disabled, while both DTR and RTS are set to be false. When transmitting or receiving data, (i.e., an event is triggered) the DTR is enabled, whereas RTS remains false. This makes sure that only DTR is involved in handshaking, not RTS. [5]

7.3.2 Communication Software Design

The communication software is written in two parts: transmission and reception. Transmission is for automatically or manually sending commands that control the moving direction of the robot. Reception is used to generate events, such as activating the camera, triggering an alarm, or indicating operation errors.

7.3.2.1 Transmission

Five buttons are generated: “Forward”, “Backward”, “LeftTurn”, “RightTurn”, and “Stop”. Clicking the mouse or pressing the keyboard can activate each of these commands. For example, when “Forward” is clicked, Chr(128) is sent for synchronization. After that, DTR is enabled. Chr(170) is then transmitted for security purposes. This means that the microcontroller only recognizes the data sequence starting with Chr(170). No other sequences can be accepted. Next, the character “f” is sent. Finally, the DTR is turned back to low. The following commands generate such a transmission:

```
MSComm1.Output = Chr(128)
Call Sleep(15)
MSComm1.DTREnable = True
Call Sleep(20)
MSComm1.Output = Chr(170)
Call Sleep(200)
MSComm1.Output = "f"
Call Sleep(200)
MSComm1.DTREnable = False
```

The Sleep( ) function is a standard Win32 API call that requires the following declaration:

```
Private Declare Sub Sleep Lib "kernel32" (ByVal
dwMilliseconds As Long)
```

To activate the keyboard inputs, two Win32 API calls are necessary. After declaration, the following two functions SetKeyState( ) and SetNumLock( ) are implemented.

```
Private Declare Function GetKeyboardState Lib
"user32" pbKeyState As Byte) As Long
Private Declare Function SetKeyboardState Lib
"user32" lppbKeyState As Byte) As Long
```

7.3.2.2 Reception

Once data is received, it is first stored into a String called rec_string. The program checks the security sequence, i.e., the beginning two bytes of the string. If the byte is Chr(170), the data will be processed. Otherwise, the data is discarded and the receiving buffer of the serial port is released.

To implement this, a simple FSM is developed. The original state is S0. If the first char is 170, it jumps from S0 to S1; otherwise it remains in S0. Once in S1, the data is forwarded for further processing.

7.4 ENTIRE PC INTERFACE

After both camera software and communication software are developed and well tested, they are integrated into one interface that controls the robot as a monitoring console. Some features are added to make the entire interface more user-friendly, but the main functions are included in the two software.

8.0 EXPENSE SHEET

The following table demonstrates the expenses our team encountered:

Table3. Items bought

Description	Quantity	Total Amount
IR LED and Phototransistors	2	$11.40
Cardboard	1	$1.15
Hex Schmitt Trigger And Ribbon Cable Connectors	1	$4.48
Ln914 Diodes	10	$1.26
1M Potentiometers	-	$1.03
RE200B human detection sensor and Fresnel lens	1	$31.85
Dazzle Digital Video Creator 50	1	$57.24
ATXR-434-ULC transceiver	2	$110.42
HT12D Decoder	1	$3.60
HT12E Encoder	1	$3.60
Tank treads and parts	1	$11.54
DB-9 90° connector	1	$6.48
Power adapter	1	$2.19
Radio Shack NiCd rechargeable Battery	1	$50.00
Total:		**$296.24**

Table4.Items borrowed

Description	Quantity	Total Amount
HC11 Microcontroller	1	Provided by the Department
TAMIYA Gearbox	1	Provided by the Department
H-Bridge	1	Provided by the Department

9.0 SUMMARY

The hardware and software components of the WatchDog were integrated successfully.
The WatchDog is able to detect the presence of humans by utilizing its Pyroelectric infrared sensors and a multi-element Fresnel lens. Once the detection is made the individual is asked to enter the password. If the password is incorrect the individual is identified as an intruder and an alert is sent to the monitoring station. A live video feed is also made accessible to the monitor. Our system consists of a distinct RF module that is able to have half duplex communication.
We have tested the WatchDog and have determined that it meets our specifications.

REFERENCES

[1] http://www.x10.com/products/x10_vk45a.htm
[2] Bantz., E. J. 29th September 2003. <http://ej.bantz.com/video>.

[3] http://support.microsoft.com/default.aspx?scid=fh;EN-US;vbNET
[4] Buchanan 1999, 185-188
[5] Buchanan, W. 1999. PC Interfacing, Communications and Windows Programming. Harlow: Addison Wesley Longman Limited. 185-188, 201, 332-339

[6]. X10 Technology Inc. 27th September 2003. http://www.x10.com/products/x10_vk45a.htm